Available online at www.sciencedirect.com
SCIENCE DIRECT
Biological Conservation 115 (2003) 55–62
BIOLOGICAL CONSERVATION
www.elsevier.com/locate/biocon

First-best, second-best, and heuristic solutions in conservation reserve site selection

Hayri Önal*

Department of Agricultural and Consumer Economics, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Received 6 March 2002; received in revised form 13 October 2002; accepted 24 November 2002

Abstract

Previous studies which dealt with the conservation reserve site selection problem used either optimization methods, specifically linear integer programming (IP), or heuristic algorithms. The trade-off between computational efficiency versus optimality has been discussed in some articles and conflicting messages were signaled. Although the problem of suboptimality was acknowledged, some authors argued that heuristics may be preferable to exact optimization because IP models are computationally complex and may not be solvable when too many reserve sites are involved. On the other hand, some studies reported that fairly large problems could be solved easily. This paper shows that although the computational complexity argument can be valid for large reserve selection problems, by properly guiding the solver and exploiting the problem structure, formal optimization can deliver second-best (near-optimal) solutions that dominate the greedy heuristic solutions.

Keywords: Reserve selection; Heuristic; Exact optimum; Second-best; Integer programming

Optimal selection of conservation reserves has been addressed in a series of papers in the past decade, e.g. Ando et al. (1998), Church et al. (1996), Csuti et al. (1997), Margules et al. (1988), Nichols and Margules (1993), Polasky et al. (2000, 2001), Pressey et al. (1993, 1996), and Vane-Wright et al. (1991). The problem is stated either as minimization of the number of reserves that >represent a given set of species, or finding a subset of the reserve sites that maximizes the number of species represented under a given budget constraint. Both versions of the reserve site selection problem can be formulated as a linear integer programming (IP) problem, as being instances of two prototype formulations known as the >set covering problem (SCP) and the >maximal covering problem (MCP) in the operations research literature (Church and ReVelle, 1974; Underhill, 1994; Camm et al., 1996).

Although both the SCP and MCP formulations and solution algorithms to solve the resulting IP problems were available for decades, many of the studies cited above have sought a solution to the reserve selection problem by using heuristic methods. A widely used approach is to select a reserve site at each step in such a way that the largest number of 'unrepresented' species will be added to the set of species jointly represented by previously chosen reserve sites. This selection rule, called the 'complementarity principle' in the biological conservation literature (Williams, 2000; Vane-Wright et al., 1991), is the basic principle of a heuristic known as the >greedy algorithm in operations research (Church and ReVelle, 1974). Complementarity-based reserve selection may involve more than one attribute, such as rarity versus species richness, and therefore it can be done in different ways in different selection problems. When another attribute with a higher priority than species richness is involved, at any step we may not always select a site that contains the largest number of unrepresented species. Pressey and Nichols (1989) show that rarity-based heuristics perform better than the simple greedy heuristic approach that selects a reserve site with maximum net contribution to the set of represented species at each step.

Cocks and Baird (1989) and Underhill (1994) demonstrated that the greedy heuristic may not necessarily yield a true optimum solution of the SCP. These arguments were iterated by Camm et al. (1996) who also discussed the advantages and disadvantages of using

* Tel.: +1-217-333-5507; fax: +1-217-333-5538.
E-mail address: h-onal@uiuc.edu (H. Önal).

0006-3207/03/$ - see front matter
doi:10.1016/S0006-3207(03)00093-4

off-the-shelf IP solvers and emphasized potential computational difficulties when solving large-scale problems. Indeed IP problems are among the hardest optimization problems, particularly when the number of integer (in this case binary) variables is too large. But, how large is too large? Church et al. (1996) addressed this issue and showed that problems with a few hundred species and reserve sites can be solved easily (within 9 s) using a commercial IP solver (*OSL*, *Optimization Subroutine Library*, a linear and linear/integer programming solver developed by IBM Corporation). However, Pressey et al. (1996) argued that in practice larger problems may arise and reported that in an empirical study involving 248 land types and 1885 reserve sites two IP solvers (*LP_SOLVE* and *LINGO*) could find the optimum solution only after running for ten days on a fast computer. In some cases these solvers simply failed to return a solution, whereas >slightly suboptimal heuristic solutions could be obtained within seconds or minutes of computation time. They concluded that whenever an exact optimum solution is practically feasible it should be pursued, yet if one needs quick solutions for large problems heuristics may be valuable and preferable because computational superiority may counterbalance the loss of optimality. On the other hand, Rodrigues and Gaston (2002) showed that even larger problems than the ones dealt with by Pressey et al. could be solved within seconds using *CPLEX* (a linear programming solver developed by ILOG).

The poor performance of the solvers used by Pressey et al. (1996) and their frustrating computational experience may be discouraging for modelers who do not have expertise in IP methods. It is obviously impractical and against common sense to tie up a fast computer for days or weeks to solve a problem for an exact optimum while a solution which deviates >slightly from the exact optimum can be obtained from less sophisticated heuristics within seconds or minutes. But, is this really the case and should we be pessimistic about formal optimization approaches when dealing with large problems? Or, should we be as optimistic as the studies by Church et al. and Rodrigues and Gaston suggest? The purpose of this paper is to shed further light on how difficult the formal optimization approach really is and present a practical approach to obtain optimum or second-best solutions of reserve selection problems. Computational evidence with various large-scale problems shows that this approach outperforms heuristics.

1. Integer programming models of reserve selection

The set covering and maximal covering formulations of the reserve site selection problem are described below. The notation used in the models is as follows: i and j denote the reserve sites and species under consideration, respectively; X_i is a binary variable where $X_i=1$ indicates that reserve i is in the network, otherwise $X_i=0$; Y_j is a binary variable where $Y_j=1$ indicates that species j is protected, otherwise $Y_j=0$; k_j is an integer ($\geqslant 1$) which denotes the minimum representation target for species j; δ_{ij} is a scalar where $\delta_{ij}=1$ indicates that reserve $i\in I$ includes species $j\in J$, otherwise $\delta_{ij}=0$; b is the available conservation budget.

The set covering formulation (SCP) that minimizes the number of selected sites while representing each species j at least k_j times is as follows:

$$\text{Minimize} \quad \sum_i X_i$$

$$\text{such that}: \quad \sum_i \delta_{ij} X_i \geqslant k_j \quad \text{for all } j \in J$$

The implicit assumption in the above model is that all species can be protected by selecting an appropriate set of reserve sites without consideration of any budget constraint. A more realistic formulation assumes a limited budget (or an upper limit on the number of reserves in the network) and maximizes the number of protected species. This formulation, called the maximal covering formulation (MCP), is described algebraically below:

$$\text{Maximize} \quad \sum_j Y_j$$

$$\text{such that}: \quad \sum_i \delta_{ij} X_i \geqslant k_j Y_j \quad \text{for all } j \in J$$

$$\sum_i c_i X_i \leqslant b$$

The standard SCP and MCP formulations assume that each species must be represented at least once, that is $k_j=1$ for all j. The formulations given here are more general and allow multiple representation targets.

Both the SCP and MCP formulations of the reserve selection problem are linear IP problems that can be solved, usually without serious difficulty, using commercial optimization software. However, solving such problems can be problematic in some cases, specifically when the number of reserve sites considered for selection is large. Pressey et al. (1996) report cases where an optimum solution could not be obtained in a reasonable solution time or could not be obtained at all. The reason for this difficulty lies in the algorithmic procedure, called the 'branch-and-bound' (B&B) algorithm, used by most IP solvers.

2. The branch-and-bound algorithm

In order to understand why an >exact optimum solution of the seemingly simple problems described above cannot be found after running a fast computer for several hours or even days, one needs to understand how the B&B algorithm works. The details of this algorithm can be found in optimization textbooks, but it will be outlined very briefly here to facilitate the discussion that will follow. The B&B algorithm is an iterative procedure based on successive partitioning of the solution set. The partitioning process imposes a bound (upper or lower) on the value of an integer variable at each step and solves a >relaxed subproblem (a linear program) that treats the >unbranched integer variables as continuous variables, starting with one variable first and gradually increasing the number of >branched integer variables, until an integer solution is found. This generates a ‘tree’ which consists of branches and nodes, where at each node a relaxed linear programming problem is solved. The number of nodes can be extremely large even for the simplest case where all integer variables are binary (specifically 2^n nodes for n binary variables). To avoid an exhaustive search of the tree, which may be practically infeasible even for moderate size problems, the algorithm generates a >bound on the optimum solution as it progresses. During the B&B iterations many integer solutions can be found and the bound is updated as improved solutions become available. If an integer solution is found or the relaxed problem solution at any node is worse than the bound, further branching is not done beyond that node. This is an extremely useful feature of the algorithm. If the user can provide a good initial bound, called the >cutoff value, this may eliminate the search over a substantial portion of the B&B tree and increase the chances for finding an optimum solution. Besides the bound, at each iteration the algorithm generates the >best possible value of the unknown integer solution and calculates the absolute and percentage deviation of the incumbent solution from that value, called the >absolute gap and >relative gap, respectively. The gap parameters, specified by the user, set the optimality tolerance and the B&B algorithm stops when an integer solution is found satisfying the tolerance level. If an exact optimum is sought with 100% confidence and there is no a priori knowledge about the optimum value, the gap parameters must be specified as zero, which allows no tolerance for suboptimality. Note that this may require an exhaustive search of the entire tree (except >fathomed branches).

Generally, the algorithm finds an integer feasible solution early in the process by using a built-in heuristic (not the greedy heuristic) and then uses the B&B iterations to improve the solution. It progresses very fast in the beginning and may even find an optimum solution of the problem in those early iterations. However, a substantial amount of iterations and processing time may be needed to >confirm that the incumbent solution is optimal. A proper optimality criterion and a good cutoff value can help tremendously to avoid numerous iterations, although this may terminate the search prematurely and yield a suboptimum solution. Commercial IP solvers offer help and report at any iteration when an improved integer solution is found. The rate of progress and relative/absolute gap levels are also reported with a specified frequency (for instance, *OSL* reports after solving every 20 nodes). This allows the user to infer how good a solution is at any iteration even if the exact optimum solution is unknown. Unlike other optimization methods, such as linear and nonlinear programming, integer programming is highly user dependent and demands guidance, but in general it responds very well to quality input.

How can we use the above ingredients of IP when solving a large SCP or MCP? For the general IP problem, it may not be possible to tell whether there can be an integer solution between the best possible and incumbent solutions. However, the SCP and MCP have a special characteristic that can be exploited, namely the objective function can take integer values only (the number of reserves or species). Therefore, if a solution has an absolute gap less than 1.0, it has to be optimal and the algorithm should be terminated. Just this manipulation may eliminate a substantial number of useless iterations and several hours of computation time when solving large problems. Second, if an integer solution is found and z_0 is the corresponding objective value, the user can interrupt the program, supply a cutoff value of z_0-1 in SCP and z_0+1 in MCP, and restart. By doing so, numerous branches and a good portion of the B&B tree can be skipped during the solution process.

3. Computational experience with large SCP and MCP models

To see the effectiveness of user intervention methods mentioned above, a number of SCP and MCP problems were generated randomly including 50–350 species and 200–3500 reserve sites. The computational experience with a state-of-the-art optimization software, *GAMS* (Brooke et al., 1992) interfaced with *OSL,* showed that both the SCP and MCP formulations can be solved to an exact optimum usually within seconds or a couple of minutes of processing time for small problems having up to 100 species and 1000 reserves. Formal optimization results were in general 10–15% better than the greedy heuristic solutions of the same problems. These findings are consistent with the computational efficiency results reported by Church et al. (1996), Pressey et al. (1996), and Rodrigues and Gaston (2002).

For moderately large problems, however, an exact optimum solution could not be obtained in many cases when the algorithm was forced to terminate after two hours of processing time. For instance, in a test problem with 200 species and 1273 reserve sites, an integer solution with 63 reserves was found after about three thousand iterations in a >cold start (i.e. without specifying a cutoff value), which took just a few seconds. Then, a second integer solution with 60 reserves was found after completing 158 thousand iterations, which took about 320 s. The absolute gaps for the two solutions reported by *OSL* were 5.9 and 2.1, respectively. The second gap value implies that the best possible integer solution cannot be less than 58, therefore the unknown true solution has to be between 58 and 60. The next two million iterations (which took about 2 h) did not report a third integer solution, while reducing the gap very little from 2.1 to only 1.4. The last gap value shows that the optimum solution has to be 59, if it is not 60. When the relative gap criterion was specified as 0.10 (*OSL*s default), however, the cold start again found the integer solution with 63 reserves quickly and stopped immediately, because the calculated relative gap (9.9%) was less than the specified tolerance level. Then the program was restarted with a cutoff value of 62, and successively with 61 and 60 reserves. In the first two restarts an integer solution yielding the given cutoff values were found in less than 6 s, while in the third restart the solution was found in about 50 s. An integer solution could not be obtained with the cutoff value of 59 in 5 min of processing time and *OSL* was forced to stop. Finally, a cutoff value of 58 was tried to see if a feasible solution exists. *OSL* reported in merely 37 s that a feasible integer solution does not exist. Thus, the best available integer programming solution with 60 reserves is either optimal or the second-best. On the other hand, the greedy heuristic solution for the problem indicated that 69 reserves should be selected to cover all the species, 15% more reserves than the second-best IP solution. Not only the number of reserves, but also the lists of selected reserves were different in the two solutions. Of those 69 reserves selected by the greedy heuristic, only 26 were included in the IP solution. The total processing time needed to do the four IP runs (including the cold start with 0.1 relative gap) that produced successively reduced integer solutions was only 70 s. Thus, a significant improvement was offered by formal optimization relative to the heuristic solution and this can be achieved at the expense of very little computation time. Furthermore, even though we cannot guarantee that the best available solution is optimal, we know that it can be at most one reserve away from the true optimum solution. The heuristic approach offers no clue regarding this matter.

The situation depicted above was not unique. Many other test problems indicated a clear comparative advantage of integer programming over the heuristic approach in terms of solution quality, and the computation time was not seriously restrictive. In none of the test problems could the greedy heuristic find an equal or smaller number of reserves or a larger coverage than the formal optimization approach even though the IP solutions were not confirmed to be optimal. It should be noted, however, that data characteristics can be important in reserve selection using heuristics (Pressey et al., 1999). The random generation of species presence data may have influenced the degree of suboptimality of heuristic solutions reported here.

The results of four selected runs with a large number of species and reserve sites are displayed in Table 1 (run no.1 through run no.4). In most cases, a cold start produced an integer solution within the first few thousand iterations, which took seconds or at most a minute of processing time, and in some cases B&B iterations generated improved intermediate integer solutions. However, in none of those five problems the final solutions could be confirmed to be optimal although one million iterations were completed when *GAMS*/*OSL* was forced to terminate. Starting from a cutoff value given by the greedy heuristic solution produced the same results. Therefore, using the heuristic approach prior to formal optimization does not offer any extra help. This is so because *OSL* and other IP solvers have their own built-in heuristics that usually generate an integer solution quickly. In all test problems considered here the initial heuristic solutions generated by *OSL* dominated the greedy heuristics. Then, the relative optimality gap was increased to 0.20 (allowing a 20% deviation from the unknown optimum instead of 1% assumed in the cold start) and improved cutoff values were provided to *OSL*, each time one less than the previous solution beginning with the first integer solution obtained from the cold start (see the >initial estimate column in Table 1). The algorithm stopped usually within seconds or minutes (at most 12 min). Improved solutions could be found in successive iterations, which took not more than 5 min altogether in three of those four cases. The last run in each case was repeated with a reduced relative gap specification (0.01), but still the solution was not confirmed to be optimal after one million iterations.

The largest problem (run no.4) included 350 species and 3111 reserve sites, which is much larger than the problem that was reported to be unsolvable in ten days by Pressey et al. (i.e. 248 species and 1885 reserve sites). This example best explains the advantage of the cut-branch-and-bound method suggested here. The first feasible integer solution with 100 reserves was obtained by *OSL* quickly, within only 63 s, in the cold start. A second solution with 91 reserves was found in the same run after about 380 thousand iterations, which took about 28 min. A further improved solution could not be found in the next 600 thousand iterations and *OSL* was stopped. Starting the cut-branch-and-bound procedure with the first integer solution (100) and reducing the cutoff value by one at a time, six more integer solutions could be found (for space reasons, only four of those solutions are reported in Table 1). Note that since the

Table 1
Comparison of the results of *GAMS/OSL* and heuristic solutions for optimum reserve site selection that covers all species

		Number of selected sites			
Run No.	Number of species/reserves	Heuristic solution	Initial estimate	IP solution[a]	Best possible solution[b]
1	250/1108	**74**	–	67	61
			66[c]	66 (46.0)	61
			65	65 (6.7)	61
			64	**64** (88.5)	**61**
2	250/2508	**78**	–	71	66
			70	70 (10.7)	66
			69	**69** (253.7)	**66**
3	300/2642	**93**	–	78	73
			77	**77** (23.0)	**73**
4	350/3111	**105**	–	100	86
			99	99 (53.4)	86
			98	98 (52.1)	86
			97	96 (10.2)	86
			95	95 (65.1)	86
			94	94 (1592.8)	86
			93	**93** (535.1)	**86**
5	250/2293	**253**	–	225 (44.7)	211
	$k_j = 3$		222	222 (38.1)	211
			220	219 (32.4)	211
			218	218 (29.6)	211
			217	217 (33.4)	211
			216	**216** (38.1)	**211**
6	250/2293	**305**	–	274 (107.)	256
	$k_j = 3$ or 15% [d]		273	273 (171.4)	256
			272	272 (1,479.2)	256
			271	**271** (323.0)	**256**

[a] In the IP solution column, the bold figures are the second-best integer solutions obtained from *GAMS/OSL*. Their optimality could not be confirmed. The figures in parentheses represent the processing times in seconds when each solution was found and the solver was forced to terminate by specifying a large relative optimality criterion (20%).

[b] The best possible solution column reports lower bounds for the unknown optimum solutions. These values are rounded to the nearest integer values larger than the actual solutions reported by *GAMS/OSL*. These values may or may not be the true solutions. Thus, in each run, the bold values in this column and the IP solution column determine the range where the unknown true optimum solution lies.

[c] The initial estimate values are the cutoff values provided to *OSL*. In the first row of each run, no cutoff value was assigned (cold start). After the first IP solution was obtained (with a 20% optimality criterion), a cutoff value that is 1 less than the IP solution was assigned. This process was repeated until an IP solution could not be found in one million iterations, where OSL was forced to stop (for the sake of space, some steps of run no. 5 are not presented here). In most cases, the IP solution was the same as the cutoff value (i.e., the first feasible solution was reported as the optimum solution—up to the 20% optimality criterion), but in some cases it was smaller.

[d] In this run, the minimum representation targets were specified differently for individual species. Specifically, for each species it is at least 3 or 15% of the number of sites including that species (rounded to integer values), whichever is larger.

first solution (100) was quite far from the best possible solution (86 reserves), a feasible integer solution could be searched in between, such as 94 reserves, skipping the restarts between 99 and 94. Just to see where the true optimum solution of the problem was and how long it would take to obtain that solution, the cold start was repeated with zero optimality criterion and a very large iterations limit (20 million). In about 15 h of runtime and solving nearly 80 thousand nodes (relaxed linear programming problems), no better solution than 91 could be obtained. After all those iterations, the absolute gap could be reduced only to 3.4 (thus increasing the best possible solution of the problem from 86 to 88), without confirming the optimality of the best available solution. This observation shows that the computational difficulty encountered by Pressey et al. (1996) may not be unique or may not be a solver-related problem. (Unpredictable performance differences may occur when using two different codes to solve the same problem, because each code has its own way of handling the branch-and-bound tree. While one solver may find a solution early in the process, the other may have to cover the entire tree for that.) On the other hand, Rodrigues and Gaston (2002) report a case where the > optimum solution of a SCP with 615 species and 1858 reserves was found in merely 2.2 s. This out-of-the-ordinary processing time may either be a very special case (due to data characteristics, occasionally either during the initial heuristic attempts or after solving the first few relaxed linear programming problems, the B&B procedure may return an exact optimum integer solution), or what they have found was actually an 'intermediate' solution (rather than the exact optimum) reported as the optimum solution by *CPLEX* if a large optimality criterion was used.

The last column in Table 1 displays the best possible integer solutions, generated by *OSL,* for those four cases. The true optimum solutions may be anywhere between those values and the best available integer solutions reported in the IP solution column (figures in bold). It is quite possible that the IP solutions are the true optimum solutions, but this cannot be guaranteed. Assuming that the worst case occurs, i.e. the last column reports the true optimal solutions, the relative gaps would be 4.9, 4.5, 5.5, and 8.1%, respectively, for the four cases. These deviations may seem significant, and indeed they are. However, they are far better than the heuristic solutions, as they suggest selecting 10, 9, 16, and 12 fewer reserves (see Table 1) which correspond to 13.5, 11.5, 17.2, and 11.4% relative improvement, respectively.

Pressey et al. (1996) report that the deviations between their heuristic and optimum solutions were slight. The deviations between the heuristic solutions and the optimum or > second-best solutions found here are not slight. The economic cost of setting aside thousands of acres of land with alternative uses and monitoring and maintenance of extra 10–15 reserves can be substantial, as also argued by Rodrigues and Gaston (2002). Implementing a reserve selection strategy based on IP solutions, that may be available within a few minutes or an hour of computation time, can save valuable conservation resources. We must not forget that the driving force in reserve selection concerns not the computational complexity or processing time, but it concerns the scarcity of economic resources.

4. Additional selection criteria

Incorporating additional selection criteria may make the matter worse for heuristic solutions and widen the suboptimality. Here I consider priority differences between species groups as an example (priority differences between reserve sites can also be analyzed similarly). Underhill (1994), Camm et al. (1996), and Church et al. (1996) all emphasized the possibility of imposing a site to be included in a reserve network or a species to be covered by fixing the value of the corresponding binary variables at one in an IP model. A different type of priority relationship may require some species to be protected first, regardless of the presence of other species in selected reserves, and only after the critical species are covered (within the budget or reserve number limitations), protection of other species can be taken into account as additional benefit. This type of restrictions can be directly incorporated in the MCP formulation or by using relatively large weights for critical species, as suggested by Church et al. (1996). Incorporating such priority relationships can be more problematic in the heuristic approach to MCP. The greedy heuristic can handle the problem in different ways. One way is at each step select a reserve site that contains at least one unrepresented critical species and represents more species (in terms of both critical and non-critical species) than alternative sites. Alternatively, at each step we may select a reserve site that covers the largest number of unrepresented critical species disregarding the total number of species (as a second rule, ties can be broken by selecting the site with the largest number of additional species). Both approaches were applied in a problem with 100 species, of which 20 were assumed to be seriously endangered and assigned absolute priority, and 445 reserve sites of which at most nine were allowed to be selected. The first approach described above covered 49 species in all, but five of the critical species were not covered because of the reserve site limitation. The second approach covered 19 of the 20 critical species and 34 species all together. Assuming that the main concern of the analysis is protection of the 20 critical species, the second solution should be preferred. The IP formulation, on the other hand, found a solution that covered all of the 20 critical species and a total of 41 species, thus dominating the better heuristic solution in terms of both the critical and non-critical species. This is not surprising because the IP solution considers the entire set of feasible combinations and chooses the best alternative, whereas the heuristic approach proceeds in a myopic way.

The standard SCP formulation used in the above problems requires that each species must be contained in at least one selected reserve, i.e. $k_j = 1$. In practice, however, multiple representation targets (occurrence of some species in more than one site) may be required to improve the long-term persistence of rare or critically endangered species. This type of requirements can be incorporated by specifying the right-hand-side coefficients of the related representation constraints as integers greater than one. Pressey et al. (1996) report that in their analysis an optimum solution could not be obtained after imposing such a requirement (a minimum percentage representation for individual land types). There is no particular reason that makes the latter problem more difficult, because the multiple-representation requirement does not alter the structure of the B&B tree. The only change occurs in the right-hand-side coefficients of relaxed linear programming problems that will be solved at each node, which would not pose any computational difficulty. However, since typically a solution of the standard SCP is not feasible for the multiple-representation problem (namely, when a species is represented only once in the SCP solution whereas the multiple representation may require $k_j > 1$), the nodes visited during the B&B iterations may be entirely different in the two cases. Therefore, we may be able to find a solution of the standard SCP (with $k_j = 1$) within a reasonable runtime, but when multiple representation is required obtaining a solution may take too long (or practically impossible), depending on what portion of the B&B tree needs to be covered. The cut-branch-and-bound procedure described in the foregoing discussion can again be used to solve such problems, as will be elaborated below.

To see the effect of multiple representation requirements, a sample problem was generated randomly including 250 species and 2293 reserve sites. The occurrences of individual species among those 2293 reserve sites varied (again randomly) between 5 and 40. The selection problem was solved first with $k_j = 1$, and then with $k_j = 2$ and $k_j = 3$, for all j. Finally, the representation targets were specified differently for different species by setting $k_j = 3$ or 15% of the total occurrence of species j, whichever is larger. Table 1 reports the results of the last two runs (labeled as run no.5 and run no.6) along with the heuristic solutions. It is possible to design different heuristic rules for this problem. In the particular application, at each step the heuristic algorithm selects a site which contains the largest number of species that are not yet represented at least k_j times in previous selections. With this rule, the heuristic approach found that 253 reserve sites would be required to represent each species at least three times (Table 1, run no.5). In contrast, the cold start IP solution required only 225 reserves, which could be reduced later to 216 in five successive runs (which took less than 5 min all together).

The best possible IP solution indicates that 211 reserves might be necessary (see Table 1, last column). Therefore, even though optimality of the second-best solution (216) is not confirmed, we know that it can be at most five reserves away from the true optimum solution (if the worst case occurs, i.e. if selecting 211 reserves is optimal). However, it is far better than the heuristic selection (37 fewer sites, which corresponds to 16% improvement). In all cases with multiple representation targets, the optimal solution could not be obtained after completing one million B&B iterations when the solver was stopped. These results suggest that when a large number of reserves and multiple representation criteria are involved, it may be generally difficult to obtain an exact optimum without interference to the IP solution process. However, as in the case of single representation problems, the cut-branch-and-bound procedure can again be used conveniently to produce a near-optimum reserve selection in a reasonably short processing time. Moreover, this approach significantly outperforms the greedy heuristic approach.

5. Conclusions

The main conclusion of this paper is this: formal optimization should be preferred to heuristic approaches, regardless of the problem size or complexity, as long as the reserve selection problem can be formulated in a linear IP framework. In general, computation time does not pose a serious obstacle when working with fairly large models. Even if an exact optimum solution is difficult to obtain, second-best (near-optimum) solutions can be found within a reasonable computation time. As demonstrated in this study, IP solvers can deliver considerably improved reserve selection alternatives (even though they may not be optimal) compared with heuristic solutions. This can be done by properly guiding the solver, such as the cut-branch-and-bound approach presented here. This approach can be used conveniently in any reserve selection problem if the objective function is integer valued (such as the number of reserves, species, or boundary edges, etc.).

Besides superior output, state-of-the-art optimization software such as *GAMS*, *AMPL*, *MPL*, etc., offer other practical advantages. They are interfaced with a long list of linear IP solvers each having comparative advantage in different problem situations and operate on different domains, including PCs, workstations and mainframe computers, without any change in the source code. This opens wide possibilities for collaborative research and allows easy access to different models developed by different research groups. These are in general difficult with custom-made heuristic programs.

It is generally perceived that IP is not suitable for modeling reserve selection problems with spatial considerations (such as proximity and connectivity of sites in a reserve network) or for incorporating genetic diversity in species conservation. Several studies have used heuristics just because of this largely false perception. Recent studies by Önal (2002), Önal and Briers (2002), and Rodrigues and Gaston (2002) showed that the SCP and MCP formulations can be extended to incorporate additional selection criteria, including spatial considerations and genetic diversity, without any serious computational problem. Thus, the real challenge is to broaden our modeling library to incorporate additional and realistic selection criteria in a linear IP framework, rather than how to deal with the computational complexity of integer programming.

References

Ando, A., Camm, J.D., Polasky, S., Solow, A., 1998. Species distribution, land values and efficient conservation. Science 279, 2126–2128.

Brooke, A., Kendrick, D., Meeraus, A., 1992. GAMS, General Algebraic Modeling System, Release 2.25, A User's Guide. GAMS Development Corp, Washington, DC.

Camm, J.D., Polasky, S., Solow, A., Csuti, B., 1996. A note on optimal algorithms for reserve site selection. Biol. Conserv. 78, 353–355.

Church, R.L., ReVelle, C., 1974. The maximum covering location problem. Papers of the Regional Science Association 32, 101–118.

Church, R.L., Stoms, D.M., Davis, F.W., 1996. Reserve selection as a maximal covering location problem. Biol. Conserv. 76, 105–112.

Cocks, K.D., Baird, I.A., 1989. Using mathematical programming to address the multiple reserve selection problem: an example from the Eyre Peninsula, South Australia. Biol. Conserv. 78, 113–130.

Csuti, B., Polasky, S., Williams, P.H., Pressey, R.L., Camm, J.D., Kershaw, M., Kiester, R.A., Downs, B., Hamilton, R., Huso, M., Sahr, K., 1997. Comparison of reserve selection algorithms using data on terrestrial vertebrates in Oregon. Biol. Conserv. 80, 83–97.

Margules, C.R., Nicholls, A.O., Pressey, R.L., 1988. Selecting networks of reserves to maximize biological diversity. Biol. Conserv. 43, 63–76.

Nichols, A.O., Margules, C.R., 1993. An upgraded reserve selection algorithm. Biol. Conserv. 64, 165–169.

Önal, H., 2003. Preservation of species and genetic diversity. American Journal of Agricultural Economics 85, 438–448.

Önal, H., Briers, R.A., 2002. Incorporating spatial criteria in optimum reserve selection. Proceedings of the Royal Soceity of London B 269, 2437–2441.

Polasky, S., Camm, J.D., Solow, A., Csuti, B., White, D., Ding, R., 2000. Choosing reserve networks with incomplete species information. Biol. Conserv. 94, 1–10.

Polasky, S., Camm, J.D., Garber-Yonts, B., 2001. Selecting biological reserves cost-effectively: an application to terrestrial vertebrate conservation in Oregon. Land Economics 77, 68–78.

Pressey, R.L., Humphries, C.J., Margules, C.R., Vane-Wright, R.I., Williams, P.H., 1993. Beyond opportunism: key principles for systematic reserve selection. Trends in Ecology and Evolution 8, 124–128.

Pressey, R.L., Nicholls, A.O., 1989. Efficiency in conservation evaluation: scoring versus iterative approaches. Biol. Conserv. 50, 199–218.

Pressey, R.L., Possingham, H.P., Margules, H.P., 1996. Optimality in reserve selection algorithms: when does it matter and how much? Biol. Conserv. 76, 259–267.

Pressey, R.L., Possingham, H.P., Logan, V.S., Day, J.R., Williams, P.H., 1999. Effects of data characteristics on the results of reserve selection algorithms. Journal of Biogeography 26, 179–191.

Rodrigues, A.S.L., Gaston, K.J., 2002. Maximising phylogenetic diversity in the selection of networks of conservation areas. Biol. Conserv. 105, 103–111.

Underhill, L.G., 1994. Optimal and suboptimal reserve selection algorithms. Biol. Conserv. 70, 85–87.

Williams, P.H., 2000. Complementarity. In: Levin, S.A. (Ed.), Encyclopedia of Biodiversity, Volume 1. Academic Press, San Diego, pp. 813–829.

Vane-Wright, R.I., Humphries, C.J., Williams, P.H., 1991. What to protect? Systematics and the agony of choice. Biol. Conserv. 55, 235–254.